SUITERA

MOReal

User Guide

The User Guide Document: Version 4.0
8-6-2020

Table of Contents

Getting started

General introduction

Electromagnetic (EM) and integrated circuit (IC) designers now work on large scale circuits that include a significant number of interconnects such as buses, transformers, inductors, and transmission lines. These circuits operate at very high frequencies, sometimes reaching hundreds of gigahertz. To accurately simulate complex circuit behavior, EM and IC designers need to include all of the resistive, capacitive, and inductive effects of the interconnects; these designers tend to do parasitic extraction using tools such as Calibre, RaptorX, and VeloceRF.

These extractors produce an enormous netlist of circuit nodes composed of resistive, capacitive, and inductive elements. This massive netlist affects solvers since the shear number of elements drastically expands the time required for analyses and simulations in addition to requiring enormous amounts of memory.

What is MOReal?

MOReal is a stand alone tool for reducing netlists composed of parasitic impedances with high accuracy and assured passivity. It provides an elegant solution for computationally expensive simulations that affect the IC design cycle by producing a reduced netlist that speeds up the simulation time by orders of magnitude while preserving the original circuit behavior and characteristics.

IC designers can integrate MOReal into their analog and digital design flows to effectively reduce netlists composed of parasitic impedances in both analog and high speed digital circuits. The tool supports different industry standard circuit formats. MOReal quickly reduces input circuits and can handle circuits with numerous ports. MOReal is also a super friendly experience as the tool only requires passing extracted circuit files and choosing preferred reduction parameters.

Who uses MOReal?

MOReal is extremely effective for IC designers who work on sensitive, high frequency, and highly dense networks of parasitic impedances. MOReal performs fast model order reduction on these netlists, producing realizable, passive, and numerically stable circuits while maintaining the behavior of the original circuits.

Analog circuit engineers who use inductors, transformers, and transmission lines operating at high frequencies can lower their design time when using the reduced netlists produced by MOReal. Engineers responsible for analog circuit layouts can speed up their simulations and reduce the required computational resources after parasitic extraction using the reduced netlists produced by MOReal.

Digital circuit designers can benefit from MOReal since SPEF files are full of instances of parasitic impedances which can be read. These files are greatly reduced while preserving high accuracy.

Key features of MOReal

High quality of reduced circuits

- MOReal produces DC clean netlists.

Circuit simulators experience difficulties during simulations and require longer simulation time because some circuits do not behave well around the DC operating point. The reduced netlists produced by MOReal have no issues when operating around DC, which lowers circuit simulation time.

- MOReal produces a highly numerically stable reduced netlist.

Numerical stability is an important property that is necessary for circuit simulators because numerically unstable systems affect the simulations by either requiring a long time to obtain the solution or fail to converge to a solution. The circuits reduced by MOReal are highly numerically stable, allowing simulators to operate seamlessly.

- MOReal can produce a physical netlist that contains only RLCK and no non-physical elements.

MOReal reduces an RLCK netlist and returns an RLCK netlist. No additional voltage or current sources are necessary. MOReal preservers the RLCK nature of the original circuits.

- MOReal can manage netlists with a large number of ports and quickly reduce these netlists.

Many model order reduction algorithms employed in other tools cannot reduce a circuit with many ports, or spend excessive time during this reduction process. Alternatively, MOReal seamlessly manages circuits with a large number of ports.

- MOReal produces guaranteed passive and realizable circuits.

Passivity is an essential parameter of model order reduction, and MOReal employs a guaranteed passive algorithm. Furthermore, MOReal produces a realizable circuit that can be interpreted by circuit simulators. MOReal produces a network of RLCK parasitic impedances that is equivalent to the original network of RLCK parasitic impedances without adding voltage or current sources.

Superior reduction

- Highly scalable software supports parallelism.

The MOReal software scales with the available hardware on the installed machine and supports parallelism.

- Fast reduction

MOReal performs a fast reduction operation due to its state-of-the-art algorithm. MOReal can decrease the design cycle for circuit designers. MOReal also has a user friendly experience since only

the circuit files require upload, and preferable parameters of the reduction can be adjusted, producing the reduced files.

- Employs a small memory footprint

MOReal is optimized for memory resources. MOReal uses minimal computational resources during the reduction process.

- Supports reduction from DC to terahertz

MOReal is not limited to a specific frequency range and can reduce circuits up to the terahertz range. MOReal guarantees the exact behavior of the netlists from DC to a user defined input frequency.

Supported features

- MOReal supports different industry standard circuit formats (SPEF, SPICE, SPECTRE).

Parasitic extraction tools produce different circuit formats to support different simulators, and MOReal can read these different circuit formats. In the case of analog circuits, the SPECTRE and SPICE formats are the most popular, while SPEF is the most popular netlist format used by digital tools. Some important notes are available to support each syntax, so it is best to check the section on "Circuit formats supported by MOReal."

- MOReal supports both analog and digital flows.

MOReal rapidly reduces the network of parasitic impedances in both digital and analog circuits with high accuracy; designers can therefore use MOReal to reduce their circuits to shorten the simulation and analysis time.

- MOReal supports two modes of reduction (wideband and narrowband).

Most physical extraction tools produce networks of parasitic impedances modeling interconnect behavior over a **wideband** of frequencies, ranging from DC (zero frequency) to a higher frequency specified by the user. Alternatively, other physical extractors produce netlists of parasitic impedances

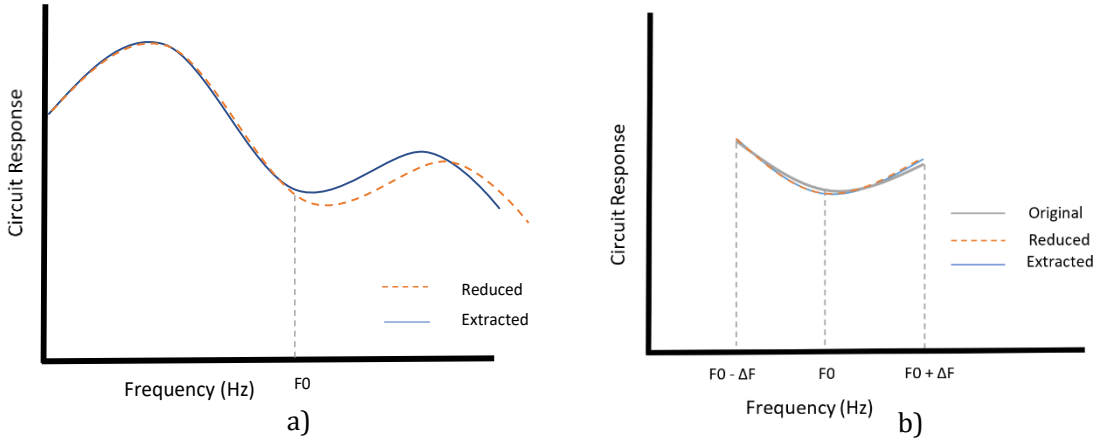

Figure 1: Frequency behavior, a) wideband approach, and b) narrowband approach

that accurately model the interconnect performance at a **narrowband** around a single, user specified, frequency, as shown in Figure 1. In addition, **narrowband** reduction can be applied to wideband extracted circuits to produce smaller reduced circuits.

- MOReal reduces RC, RCC, RLC, and RLCK circuits.

Physical extractors provide different options for the extracted circuit to be either RC, RCC, RLC, or RLCK. MOReal automatically detects and reduces the circuit, either RC, RCC, RLC, or RLCK.

- Compatible with most analog and digital EDA tools.

The output reduced circuit format is compatible with different circuit simulators such as Cadence Virtuoso, Multisim, and PrimeTime.

- User friendly interface
    - Input your netlist
    - Select reduction parameters
    - Obtain reduced netlist

About this manual

The goal of this manual is to describe how to use MOReal. This manual serves as a reference guide for users with information covering the supported netlist formats and the best approach for utilizing popular extractors with MOReal.

Feedback

We are always interested in collecting feedback on your MOReal experience; therefore, all comments and suggestions related to this document or MOReal in general are sincerely welcomed. Please send your comments and suggestions to [***support@suiteratech.com***].

Using MOReal within the Suitera website

1. Enter your files
2. Define (Subckt name & ground name) + reduction parameters
3. Submit

Frequently asked questions about MOReal

1. What is model order reduction? And why is it needed?

The market always pushes towards high performance integrated circuits (ICs) that can perform multiple operations at once; therefore, industry constructs complicated ICs operating at very high frequencies to support this need within the marketplace. However, extracting the parasitic impedances of these ICs produces an enormous circuit composed of a large number of elements, making it difficult and often impossible to simulate these huge circuits to analyze delay, noise, power consumption, and other essential design criteria.

The reduction of an enormous circuit into a small circuit while preserving all of its properties and characteristics is the function of "**model order reduction (MOR)**." MOReal is a MOR-based tool that reduces large circuits composed of parasitic impedances into a much smaller circuit ready for post-layout simulations and analyses. As an example, consider Figure 2.

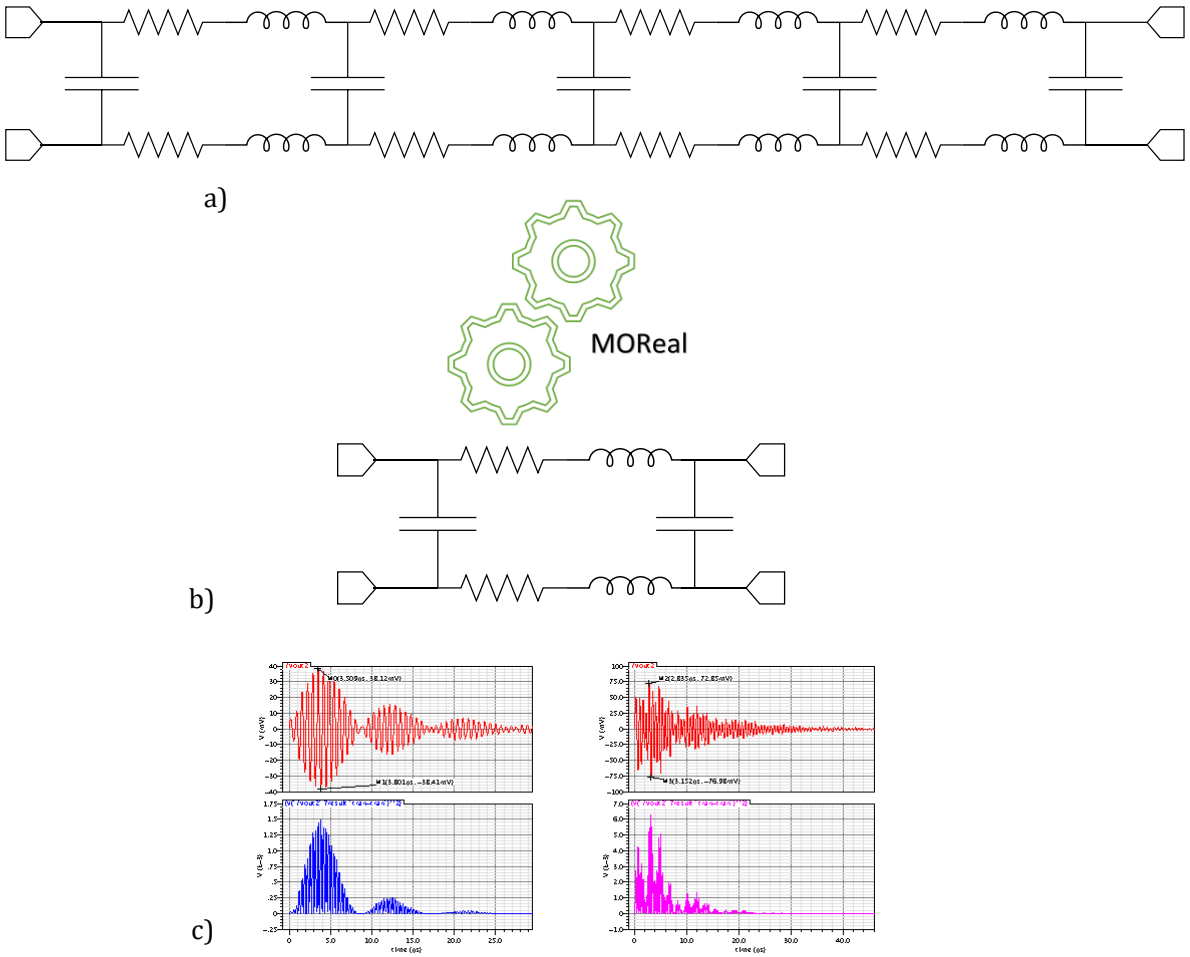

Figure 2: MOReal operation, a) large extracted circuit (original), b) reduced RLCK circuit with fewer parasitic elements, and c) simulation of the reduced circuit.

2. What is MOReal?

MOReal is a state-of-the-art model order reduction tool for the reduction of extracted parasitic impedances. With its exceptional capabilities, IC designers and electromagnetic experts can use MOReal to rapidly reduce their netlists composed of a massive number of parasitic impedances, allowing the designer to perform different analyses and simulations much faster than before using the reduced circuit produced by MOReal.

3. Is MOReal a modeling tool?

No, MOReal does not model or extract parasitic impedances. MOReal reduces netlists of parasitic impedances after performing a parasitic impedance extraction process using tools such as Calibre, RaptorX, Quantus, or Star RC. MOReal applies a novel mathematical model order reduction algorithm that accurately converts a massive network of linear parasitic impedances into a much smaller netlist while retaining the behavior and characteristics of the original circuit.

4. What are the technology nodes supported by MOReal?

MOReal manages parasitic impedances produced from any technology nodes. MOReal is independent of the technology node since the underlying algorithm operates after the extraction process is complete.

5. Does MOReal require any PDK files or layout files to run?

No, MOReal does not require any PDK files or layout files to run. MOReal only requires the circuit files describing the netlist of parasitic impedances produced by an extraction tool.

6. What are the input files required by MOReal?

After completing the layout process and performing parasitic extraction, MOReal only requires the parasitic files produced by the extraction tool from your design flow, as explained in the "MOReal User Manual" section.

7. What is the frequency range supported by MOReal?

MOReal reduces circuits from DC up to the terahertz range. MOReal supports different model order reduction modes based on the frequency specified by the user. MOReal accepts S-parameter analyses as an approach for ensuring the accuracy of the reduced netlists. Some extraction tools produce reduced circuits that either work from DC, zero frequency, to a high operating frequency, which is called wideband extraction. Other circuits only operate at one frequency, which is

narrowband extraction. MOReal supports either type of extracted circuits, wideband and narrowband.

8. Is it acceptable to use an already reduced circuit with MOReal? Why not provide MOReal with a reduced circuit for further reduction?

It is not advisable to reduce an already reduced netlist because the earlier reduction process limits the capabilities of MOReal. Typical MOR tools do not support an RLCK realization and produce a non-physical netlist that can contain voltage and current controlled instances such as voltage or current controlled sources. MOReal reduces the physical instances, resistors, capacitors, inductors, and mutual inductors (R, L, C, K), into an equivalent system of the same type of elements.

9. Is MOReal compatible with other EDA tools?

Yes, MOReal produces an output file written in SPICE, SPECTRE, or SPEF format that can be read by analog and digital circuit simulators such as Cadence, Multisim, and HSPICE.

10. Where should MOReal be integrated into an analog IC design flow?

A typical analog design flow:

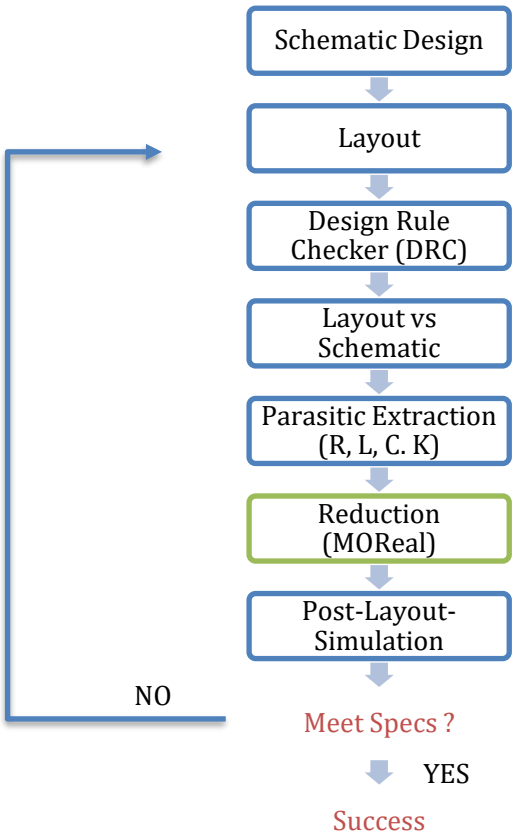

Figure 3: MOReal within an analog design flow

11. Where can MOReal be integrated within a digital IC design flow?

In the digital design process, parasitic extraction is essential for performing static timing analysis to determine the delay of the network, do noise and crosstalk analyses, perform signal integrity checks, and do logic simulations, IR analyses, and substrate noise analyses.

A typical digital design flow:

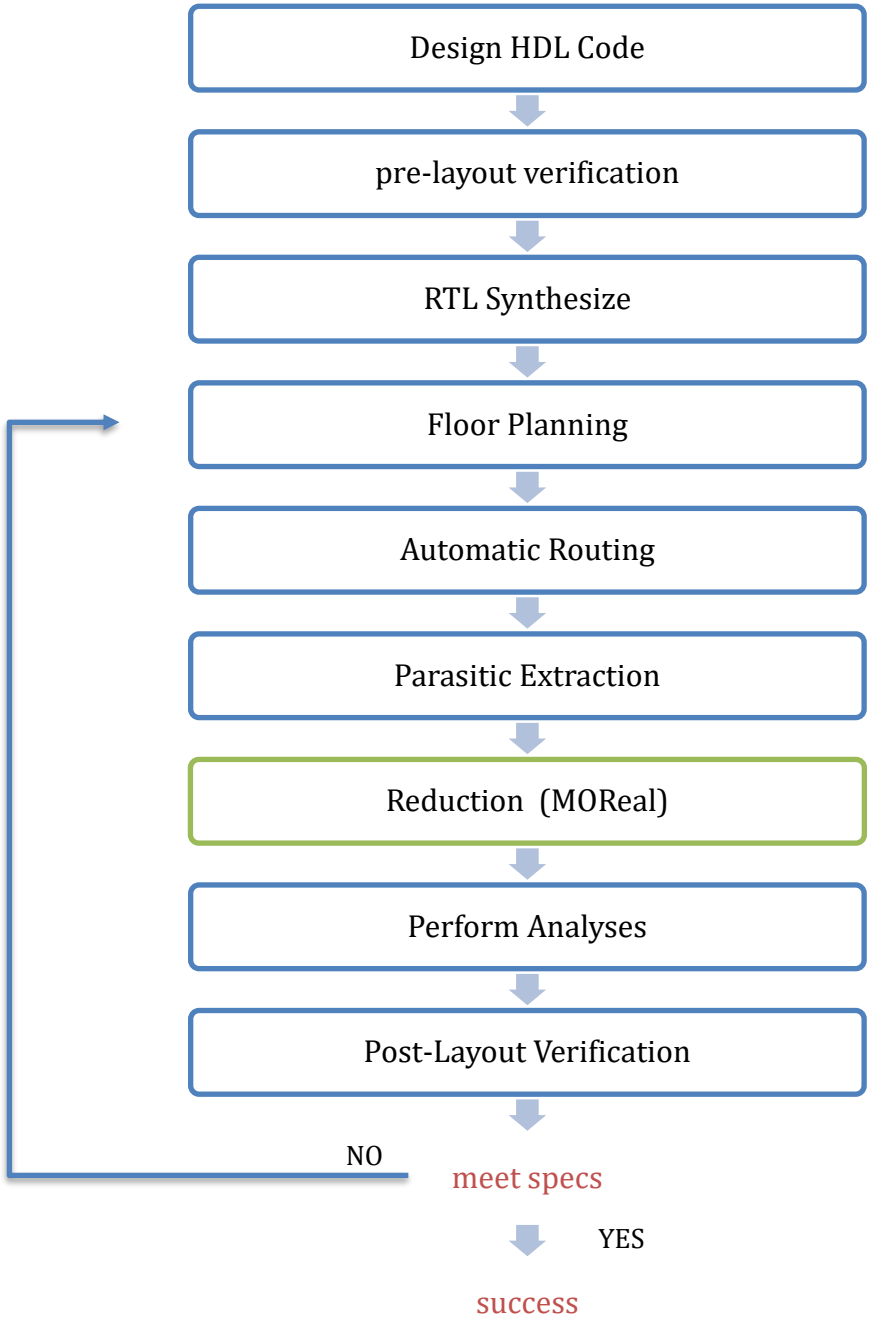

Figure 4: MOReal within digital design flow

12. How can the MOReal output simulate faster?

MOReal reduces not only nodes and elements of the input circuits but also produces a circuit with enhanced conditioning of the reduced circuit. The importance of enhanced conditioning on decreasing the time required by the iterative solvers is illustrated in Figure 5; therefore, the circuits reduced by MOReal converge rapidly on circuit simulators.

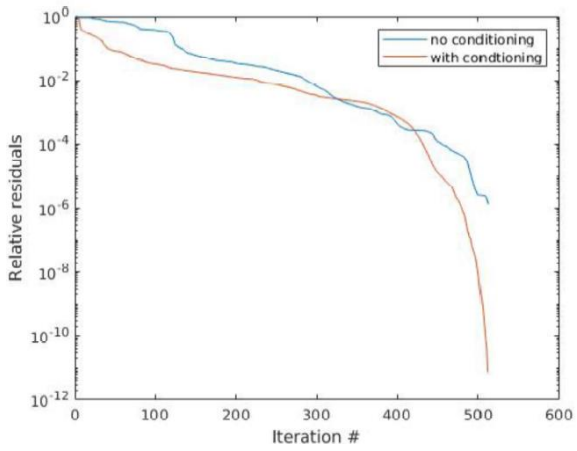

Figure 5: Systems with enhanced conditioning performs better than systems with no conditioning

13. What is the extraction process? And why is it essential?

After completing the layout of an analog or a digital circuit, engineers perform design rule checks (DRC), checking layout vs. schematic (LVS), and model the parasitic impedances of the interconnects. The process of modeling the electromagnetic effects of the interconnects is called **extraction.**

Extraction tools model these parasitic effects by including linear elements such as resistors, capacitors, inductors, and mutual inductors. These parasitic impedances significantly affect the delay, energy consumption, power distribution, and introduce noise that affects the reliability of the overall system.

Circuit formats supported by MOReal

Introduction

The supported netlist formats for MOReal are described in this section. Information about the parasitic elements recognized by MOReal is also discussed.

Supported formats: SPECTRE, SPICE, SPEF

The analog flow in MOReal supports SPECTRE, SPICE, and SPEF formats.

MOReal reads netlist files produced directly from the parasitic extractor even if active devices or numerous sub-circuit blocks are included. MOReal only considers physical reducible parasitic elements (resistors, capacitors, inductors, and mutual inductors).

General recommendation

MOReal strives to support all commands and features inside SPECTRE and SPICE syntaxes. These netlist formats are quite complex, and several commands are not interpreted by MOReal, such as defining parameters, plotting, and running analyses.

SPICE syntax support

MOReal supports SPICE3 version 3f3.

MOReal generally parses other SPICE netlist formats such as HSPICE if these formats do not contain complex commands or operations. However, the SPICE3 rules are applied to these files. Internal nodes connected to active devices are preserved as ports and are not reduced by MOReal. Active devices among other parasitic elements are retained and included in the output file.

MOReal reads netlists in different SPICE files using the ".include" command. The command is required by the user to enter a netlist file that contains the top sub-circuit in the parameters of the reduction process. The parser reads the other files.

The following syntax restrictions apply:

1. The following characters cannot be used inside the name of instances, sub-circuits or nodes.
   ‘ , =) (
2. Nodes or devices of ".NODESET", ".IC", ".DCVOLT", ".MEASURE", ".PROBE", ".PLOT", ".PRINT" and ".GRAPH" are not processed by the MOReal parser.

SPECTRE syntax support

MOReal supports SPECTRE Version 5.0.0 syntax

MOReal can read files with a (.scs) extension. The netlist file is written in SPECTRE format and has an extension different from (.scs). The user needs to specify that the syntax follows SPECTRE in the configuration section.

MOReal reads netlists written in different SPECTRE files using the "include" command. The user enters the netlist file that contains the top sub-circuit. The parser reads the other netlist files.

Internal nodes connected to the active devices are preserved as ports. These nodes are not reduced by MOReal. The active elements in the netlist file are retained and returned in the output file for the user.

The following syntax restrictions apply:

1. MOReal does not support these SPECTRE commands: [alter, altergroup, paramtest, assert]
2. The following characters cannot be used inside the names of instances, sub-circuit, or nodes.
   ‘ “ =) (} { > <

Recognition of reducible components

MOReal reduces physical parasitic instances. Distinguishing these reducible parasitic instances, among other elements in the netlist file, is important. The reducible parasitic instances are either a resistor, capacitor, inductor, or mutual inductor within the netlist file. Every syntax has a unique way of writing and describing each reducible parasitic instance. It is, therefore, essential to correctly enter the netlist format at the configuration input.

The following two sections describe the syntax of the reducible instances in SPECTRE and SPICE. Additional examples are also provided.

Reducible parasitic instances in SPICE

In SPICE syntax, each instance inside a circuit is specified by an element line that includes the following:

1. Instance name
2. Instance nodes
3. Instance value

In general, all instance names that start with (R, C, L, K) are recognized as reducible instances.

The general format of the reducible parasitic instances is listed in Table 1.

Table 1: Syntax of reducible parasitic instances in SPICE

Instance Type	General Form
Resistor	Rxxxxx N+ N- value
Capacitor	Cxxxxx N+ N- value
Inductor	Lxxxxx N+ N- value
Mutual Inductor	Kxxxxx L+ L- value

The default recognition behavior is listed in the following table.

Table 2: RLCK instances in SPICE

Instance Type	Required condition	Example
Resistor	Instance name needs to begin with the letter 'R' or 'r.'	R1 Rload
Capacitor	Instance name needs to begin with the letter 'C' or 'c.'	C100 C_cap1
Inductor	Instance name needs to begin with the letter 'L' or 'l.'	L10 L_feedback
Mutual Inductor	Instance name needs to begin with the letter 'K' or 'k.'	K1 K_L1_L2

Examples of the reducible parasitic instances:

- R1 node1 node2 1K
- C_load 100 0 1n
- L_shunt 20 25 1u
- K1 L1 L2 0.1

Important notes:

1. All of the resistors, capacitors, and inductors only have two nodes.
2. The value of the resistor, capacitor, inductor, and mutual inductor is described as a number.
3. Parasitic resistors with a negative value are supported.
4. The parser supports multiple reducible parasitic instances. Reducible instances are parallel for (m) times
   Ex. R1 5 1 30K m=3. (Three resistors connected in parallel between nodes (5, 1))
5. The two inductors in the K-instance, mutual inductor, can be local or global inductors.
6. The two inductors in the K-instance must be included in the file. If not, the parser will exit the system.
7. The value of the resistor, capacitor, inductor, and mutual inductor cannot be a formula.
   Ex: L1 n1 n2 ((100/50) + sqrt (130)) → is not supported.
8. Parasitic instances that are a function of temperature or other parameters are not supported.

All reducible parasitic instances will be eliminated or changed during the reduction process. Therefore, parasitic instances declared inside these commands do not exist after the reduction process.

[". IC", ".plot" ,".print", ".probe", ".measure"]

Non-reducible instances in SPICE

Other non-reducible instances in the netlist file are identified and returned to the user in an output file. In general, non-reducible instances are elements that are not resistors, capacitors, inductors, or mutual inductors. Non-reducible instances include voltage sources, current sources, transistors, and diodes. When MOReal encounters voltage and current sources inside a sub-circuit, MOReal terminates since these sources are indications of non-physical elements.

In a particular scenario, some users may prefer to hide a sub-circuit from the reduction tool, which can be achieved by instantiating a sub-circuit of a master or reference name while the sub-circuit declaration is not written in the file. The SPICE parser considers this instantiation as a non-reducible instance and specifies the nodes as ports. The output file will contain this instance without changing the reference name for use by the user.

There are also non-reducible instances that are defined in the SPICE syntax style. The following table lists these non-reducible instances supported by MOReal.

Table 3: Non-reducible instances

Instance type	General Form
MOSFET	MXXXXXXX ND NG NS NB MNAME <L=VAL> <W=VAL> + <AD=VAL> <AS=VAL> <PD=VAL> <PS=VAL> <NRD=VAL> + <NRS=VAL> <OFF> <IC=VDS, VGS, VBS> <TEMP=T>
Junction Diodes	DXXXXXXX N+ N- MNAME <AREA> <OFF> <IC=VD> + <TEMP=T>
Bipolar Junction Transistors (BJTs)	QXXXXXXX NC NB NE <NS> MNAME <AREA> <OFF> + <IC=VBE, VCE> <TEMP=T>
Junction Field Effect Transistor	JXXXXXXX ND NG NS MNAME <AREA> <OFF> + <IC=VDS, VGS> <TEMP=T>
MESFET	ZXXXXXXX ND NG NS MNAME
Voltage Sources	VXXXXXXX N+ N- <<DC> DC/TRAN VALUE> + <AC <ACMAG <ACPHASE>>>
Current Source	IXXXXXXX N+ N- <<DC> DC/TRAN VALUE> + <AC <ACMAG <ACPHASE>>>
Voltage-Controlled Current Source	GXXXXXXX N+ N- NC+ NC- VALUE
Voltage-Controlled Voltage Source	EXXXXXXX N+ N- NC+ NC- VALUE
Current-Controlled Current Source	FXXXXXXX N+ N- NC+ NC- VALUE
Current-Controlled Voltage Source	HXXXXXXX N+ N- NC+ NC- VALUE

Important notes:

1. The nodes connected to the non-reducible instances are treated as ports to be retained. However, the name of these instances may change since the output file is flattened. Therefore, if these special nodes are not in the top sub-circuit, the names will be changed in the output file after reduction.
2. The parameters of the non-reducible instances are returned without any changes.
3. The first parameter value needs to be preceded by "=" except for sources.

Examples of non-reducible instances:

- `M1 2 9 3 0 MOD1 L=10U W=5U AD=100P AS=100P PD=40U PS=40U`
- `M1 24 2 0 20 TYPE1`
- `DBRIDGE 2 10 DIODE1`
- `Q50A 11 26 4 20 MOD1`
- `J1 7 2 3 JM1`
- `Z1 7 2 3 ZM1`
- VCC 10 0 DC 6
- ISRC 23 21 AC 0.333 45.0 SFFM (0 1 10K 5 1K)
- Gb2_1_second 1 0 10 0 2.825240e+10
- Ereal1_1_1 23 21 12 0 4.453467e+13
- Hinf1_1 5 23 VCC 4.171863e+00
- Gb2_1_second 1 0 10 0 2.825240e+10

Reducible parasitic instances in SPECTRE

In SPECTRE syntax, each instance within a circuit is specified by an element line that includes the following:

1. Instance name
2. Instance nodes
3. Reference model
4. Instance value

The instance format should be as follows:

[instance name] (node1 node2 … nodeN) [reference name] … [param1 = value1] … [paramN = valueN]

In general, the instance reference model is equal to one of these names ("resistor"," capacitor"," inductor") to be recognized by the parser as a reducible parasitic instance.

The general format of the reducible parasitic elements:

Table 4: Reducible parasitic instances in SPECTRE

Instance Type	General Form
Resistor	xxxxx (N+ N-) resistor r=value (m=value)
Capacitor	xxxxx (N+ N-) capacitor r=value (m=value)
Inductor	xxxxx (N+ N-) inductor l=value (m=value)

The definition of a mutual inductor is different than the previous format and is specified as follows:

[instance name] mutual_inductor coupling=[value] ind1=[value] ind2=[value]

Examples of the reducible parasitic instance:

- r191 (105 106) resistor r=1.2
- C_load (106 0) capacitor c=0.0149406f
- l189 (222 149) inductor l=18.3139f
- m_1 mutual_inductor coupling=0.064 ind1=x_PM.l1420 ind2=x_PM.l521

Important notes:

1. All resistors, capacitors, and inductors only have two nodes.
2. The value of the resistor, capacitor, inductor, and mutual inductor is written as a number.
3. Negative parasitic resistors are supported.
4. The parser supports multiple reducible parasitic instances. This statement means reducible instances are parallel for (m) times.
   Ex. R1 (5 1) resistor r=30K m=3.
   (Three resistors connected in parallel between nodes (5, 1))
5. The two inductors in the K-instance, a mutual inductor, can be a local or global inductor.
6. The two inductors mentioned in the K-instance need to be included in the file. If not, the parser will exit the program.
7. The value of the resistor, capacitor, inductor, and mutual inductor cannot be a formula.
   Ex: l189 (222 149) inductor l= ((100/50) + sqrt (130)) → is not supported
8. Parasitic instances that are a function of temperature or other parameters are not supported.

All reducible parasitic instances will be eliminated or changed during the reduction process. Therefore, the parasitic instances declared inside these commands will not exist after reduction.

[". IC", "save", "nodeset" …]

Non-reducible instances in SPECTRE

Other non-reducible instances in the netlist file are identified and returned to the user in the output file. In general, non-reducible instances are all elements that are not resistors, capacitors, inductors, or mutual inductors. Non-reducible instances include voltage sources, current sources, transistors, and diodes. When MOReal encounters voltage and current sources inside the sub-circuit, MOReal terminates since these sources are indications of a non-physical element.

The instance format should be as follows:

[instance name] (node1 node2 … nodeN) [reference name] … [param1 = value1] … [paramN = valueN]

The previous format applies to all instances inside Spectre. The parser always checks for the name following the parentheses to extract the reference name. If the name does not equal to "resistor," "capacitor," or "inductor," the instance is not a reducible parasitic instance. The parser saves the reference name, ensuring not to match any of the declared sub-circuits and returns the output file to the user.

Important notes:

1. The nodes connected to the non-reducible instances are considered as ports to be retained. The name of the instance may, however, change if not in the top sub-circuit while reducing multiple layers of hierarchy.
2. The parameters of the non-reducible instances are returned without any changes.

Examples of non-reducible instances:

- M1 (2 9 3 0) MOD1 L=10U W=5U AD=100P AS=100P PD=40U PS=40U
- M1 (24 2 0 20) TYPE1
- DBRIDGE (2 10) DIODE1
- vvs (pos gnd) ccvs rm=1 probe=v1 m=1
- q1 (vcc net3 minus) npn_mod region=fwd area=1 m=1
- d0 (dp dn) pdiode l=3e-4 w=2.5e-4 area=1
- nch1 (1 2 0 0) nchmod1 l=2u w=15u ad=60p as=37.5p pd=23u ps=6u
- i1 (in 0) isource dc=0 type=pulse delay=10n val0=0 val1=500u period=500n rise=1n fall=1n width=250n
- vcs (pos gnd) cccs gain=2.5 probe=v1 m=1

Unsupported operations

1. Parameters statement (Circuit and Sub-circuit Parameters)
   parameters param=value param=value …
   parameters p1=1 p2=2
2. Parameter inheritance
3. Parameter referencing
4. Altering/sweeping parameters
5. Expressions, for example, behavioral expressions
6. Operators (algebraic and logic)
7. Algebraic and trigonometric functions
8. No built-in constants

Using MOReal with parasitic extractors

Mentor Calibre, Ansys VeloceRF, and RaptorX tools (previously Helic tools) enable parasitic extraction of interconnects such as buses, inductors, and transformers. Helic produces netlists that are extremely large and dense with parasitic impedances. It is preferable to use an extracted netlist without additional reduction performed by the extractor to exploit the complete reduction capabilities of MOReal. It is, therefore, recommended to produce the original circuit from the physical extractor without reduction for later reduction by MOReal.

ANSYS VeloceRF and RaptorX

The proper adjustments described in the following chapter guides the user to produce an original circuit from ANSYS VeloceRF and RaptorX.

Running Helic central

The tools are integrated with simulators such as Cadence Virtuoso. For example, a tab named "HelicCentral" exists inside the virtuoso window. After clicking on "Enable," the Helic central window appears, as shown in Figure 6.

Click on the "Acquire License" button required for VeloceRF. The "VeloceRF" button is now activated.
Click on the "Acquire License" button required for RaptorX. The "RaptorX" button is now activated.

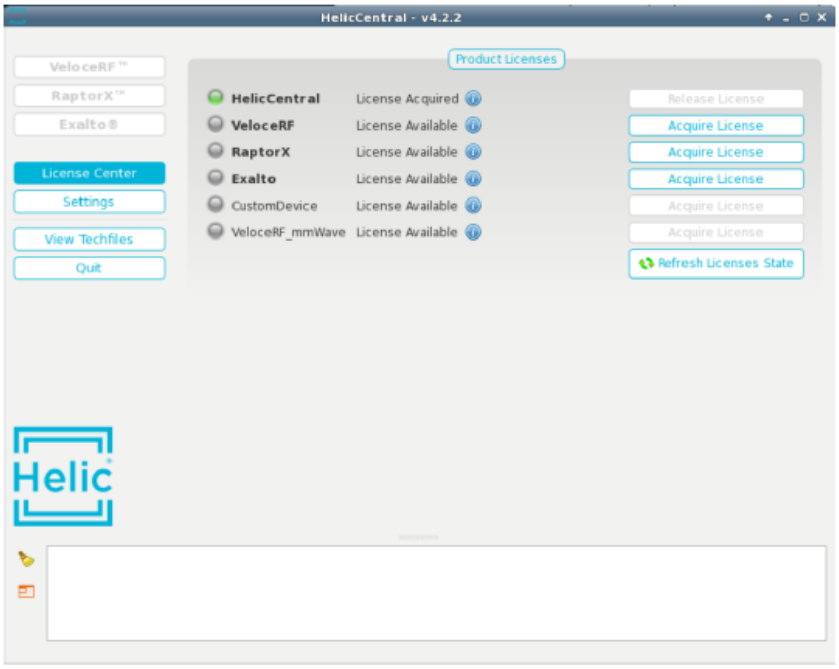

Figure 6: Helic central window to acquire the license

Adjusting settings for physical extraction

Click on the “Settings” button. A window, as shown in Figure 7, will appear.

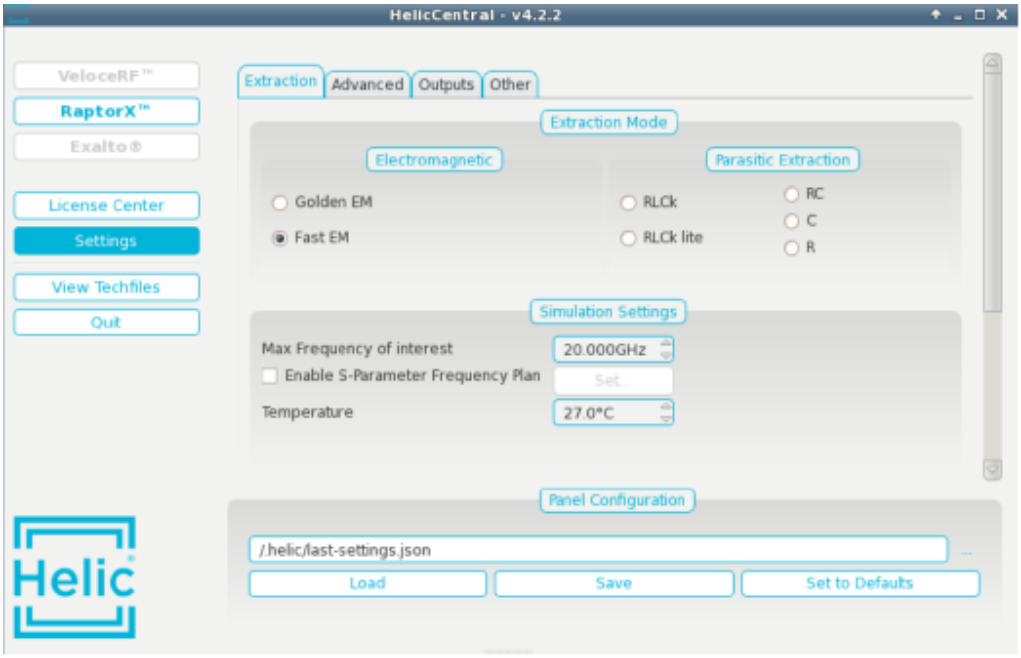

Figure 7: Helic central settings - extraction tab

In the “Extraction” tab, choose the preferred mode of extraction and the maximum frequency of extraction. In the case of RLCK extraction, the user is encouraged to select one of the following two modes according to the maximum frequency of interest.

- Extraction mode:
    - Golden EM is the most accurate model of RLCK extraction and is accurate up to 100 GHz.
    - Fast EM mode produces an RLCK impedance that is less dense than the Golden EM mode and is accurate up to 60 GHz.
- Maximum frequency of interest:
    - The maximum frequency of interest required by the user is entered and should be higher than 1 GHz.

After clicking on the “Advanced” tab, a window similar to Figure 8 appears.

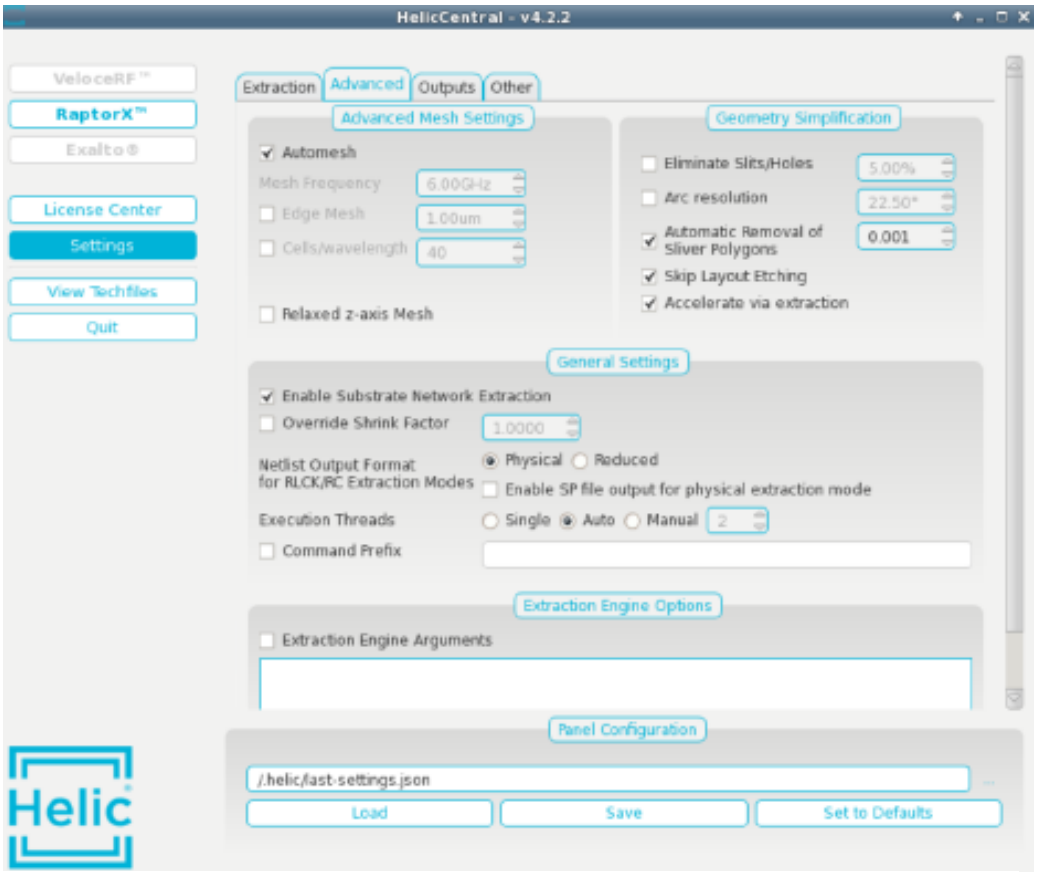

Figure 8: Helic central settings - advanced tab

In the “General Settings” section, select “Physical.”

In the “Extraction Engine Options” section, write the following commands:

```
enable-netlist-reduction (=0)

export-full-netlist (=1)
```

After clicking on the “Outputs” tab, a window similar to Figure 9 appears.

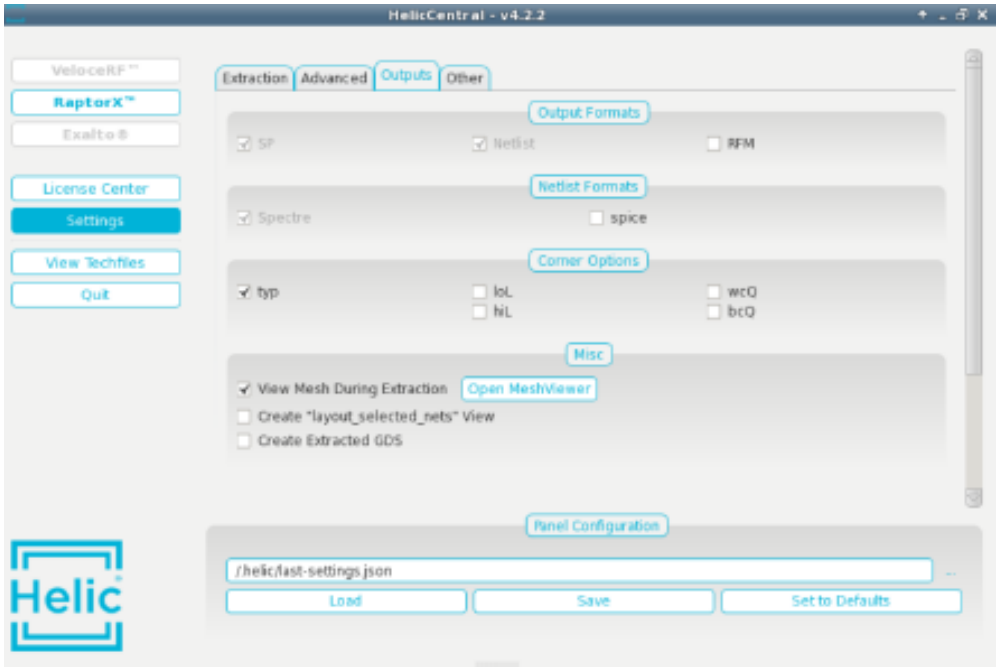

Figure 9: Central settings - outputs tab

In the “Netlist Formats” section, select the netlist to be written into SPICE format other than the default SPECTRE format.

Click on the “Save” button below to save these settings for later use.

Running VeloceRF

After adjusting the settings as previously mentioned, regular operation of synthesizing inductors, transformers, and transmission lines can be processed. The resulting sub-circuit or netlist will be in the original form ready for MOReal.

Running RaptorX

After adjusting the settings as previously mentioned, regular operation of RaptorX is applied, such as extraction of layouts and different GDS files. The resulting sub-circuit or netlist will be in the original form ready for MOReal.

Mentor Graphics Calibre

The following section provides users with the proper adjustments in the settings to produce an original circuit from Calibre.

Running Calibre

Shown below is an example of the Calibre tools integrated with Cadence Layout. In layoutXL, choose Calibre and choose PEX. As shown in Figure 10, click on the "Rules" button and input the rules file. As shown in Figure 11, click on the "Inputs" button and check export from layout viewer in the "Layout" tab.

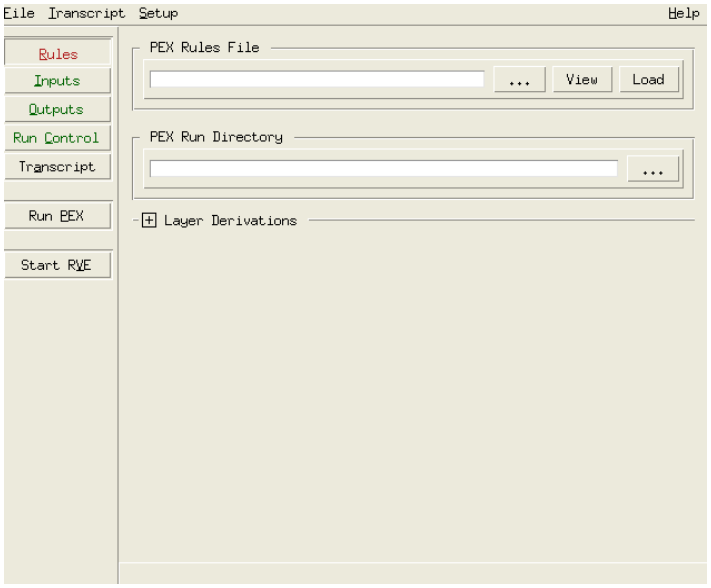

Figure 10: The rules option inside Calibre PEX.

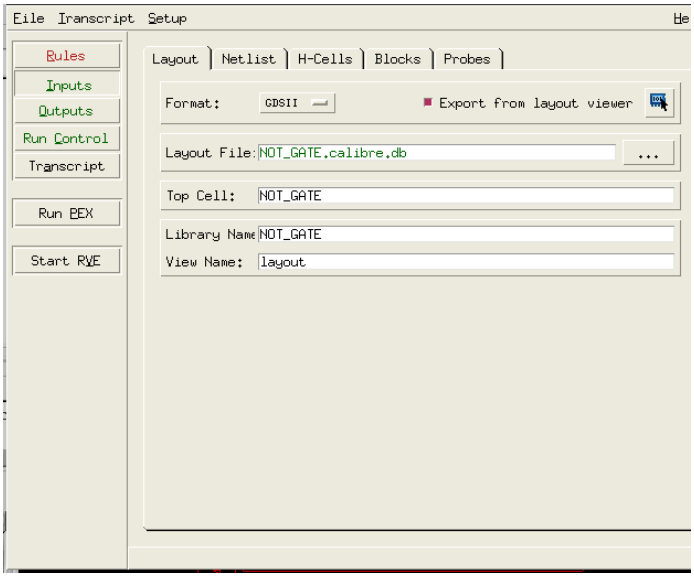

Figure 11: PEX Calibre- inputs "layout" tab

Click on the “Netlist” tab; check the export from the schematic viewer if you see an open schematic viewer, as shown in Figure 12.

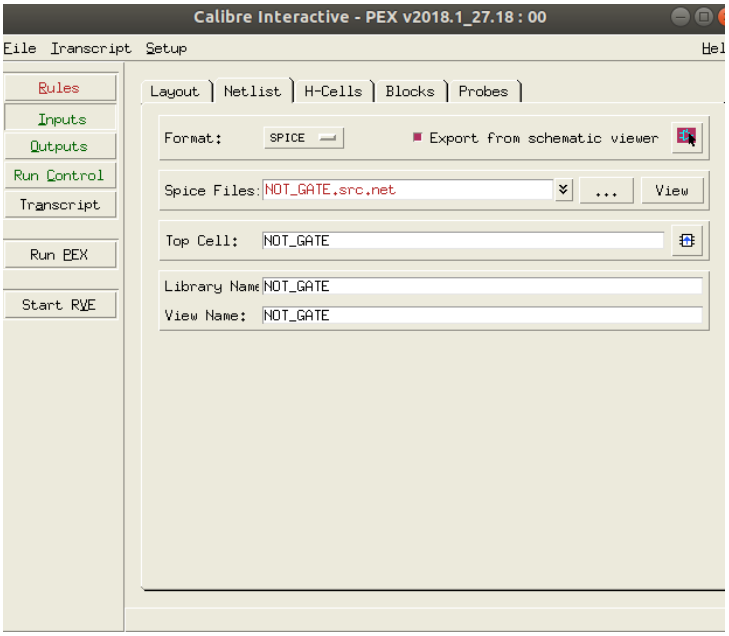

Figure 12: PEX Calibre- inputs-Netlist tab

As shown in Figure 13, click on the “Outputs” button.

In the extraction mode, choose the type suitable for your application. Choose the specified output syntax format to be either SPECTRE or HSpice.

Select the output file directory to review the results

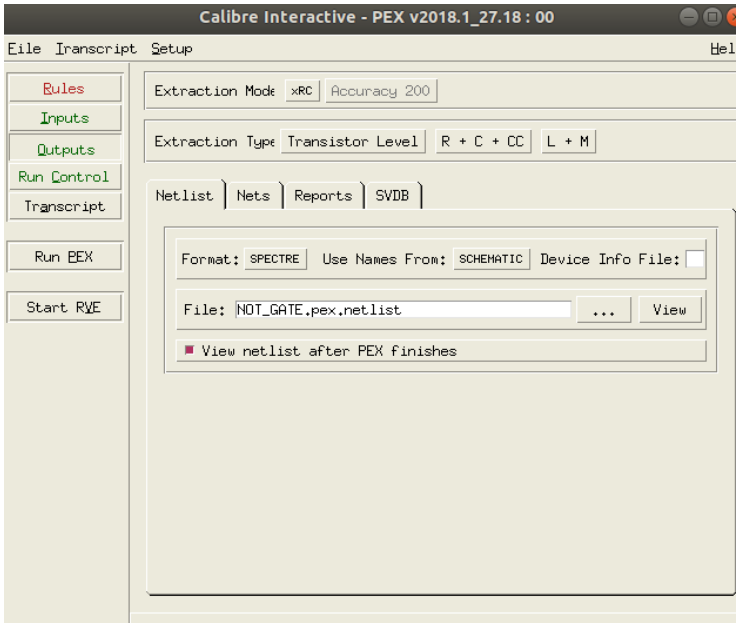

Figure 13: Outputs tab

In the case of inductance extraction, perform the following steps:

- Click on the “PEX Options” as shown in Figure 14.
- In the inductance tab, choose the extraction mode to be PEEC, choose the operating frequency. Please note that “PEEC” is for the narrowband extraction mode.
- Choose the same net mutual inductance to increase accuracy.

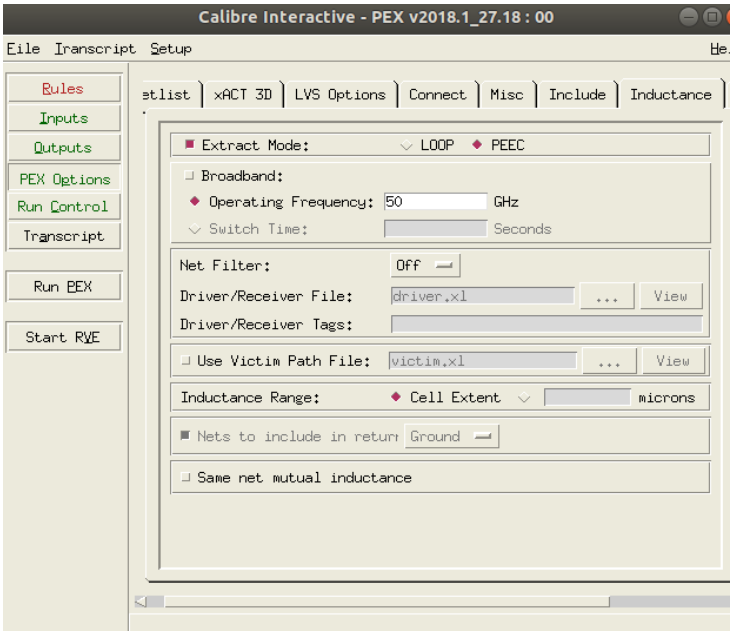

Figure 14: PEX Calibre- PEX Option

As shown in Figure 15, click on the “PEX Options,” inside the “Netlist” tab, uncheck all Reduction boxes, as shown in the following figure.

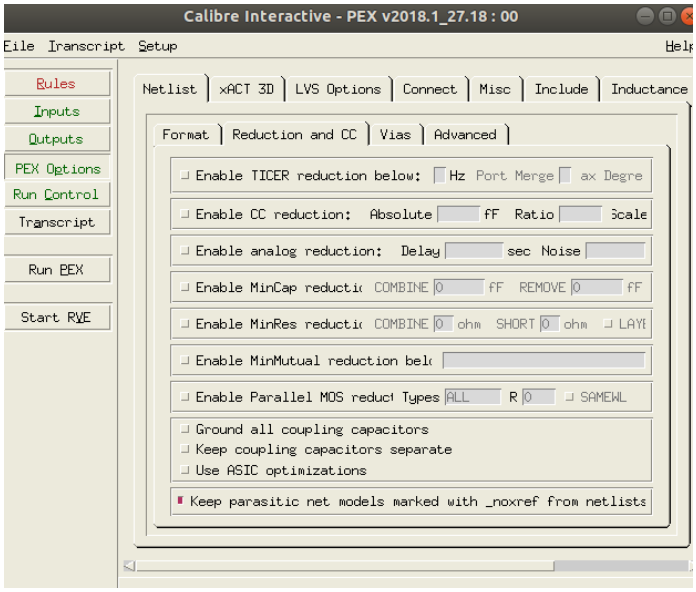

Figure 15: PEX Options -Netlists Tab

Running Calibre PEX

Click on “Run PEX,” and Calibre starts working. In the output directory, three files are generated. All three files are to be passed to MOReal. Calibre opens the top file. It is essential to specify the top file for MOReal as well as the name of the top cell.

List of figures

List of tables

References

1. **SPICE3 Version 3f3 User's Manual,**
   http://fides.fe.unilj.si/spice/download/spice3_manual.pdf
2. **Spectre Circuit Simulator User Guide, Product Version 5.0**
   https://www.ee.columbia.edu/~harish/uploads/2/6/9/2/26925901/spectre_reference.pdf
3. **VeloceRF, Inductor, Transformer, and Transmission Line Synthesize and Modeling, Manual_version4.2_rev1.4.**
4. **RaptorX, Pre-LVS Electromagnetic Modeling, User Manual Version 4.2**.
5. **Calibre Verification User's Manual.**